ARTYKUŁY

ZAGADNIENIA FILOZOFICZNE
W NAUCE
XXV / 1999, s. 76–81*

Adam OLSZEWSKI

O ROLI TEZY CHURCHA W DOWODZIE PEWNEGO TWIERDZENIA

Zadaniem niniejszego artykułu jest zdanie sprawy z matematycznej roli Tezy Churcha (w skrócie TC) w dowodzie twierdzenia o nierozstrzygalności logiki pierwszego rzędu. Ma on mieć charakter sprawozdania z wyników już znanych. Cel jaki sobie stawiam to pokazanie, w szczególności filozofom logiki, wagi filozoficznej problemu zwanego Tezą Churcha (TC).

Punktem wyjścia są poglądy Hilberta wraz z jego tzw. dziesiątym problemem, zagadnieniem, które wielce zainspirowało matematyków do odpowiedzi na pytanie, czy istnieje algorytm, który pozwoliłby w skończonej liczbie kroków rozstrzygnąć, czy wielomian w o współczynnikach całkowitych ma zero w zbiorze liczb całkowitych[1]. Następnym problemem, również postawionym przez Hilberta, był tzw. Entscheidungsproblem. Chodziło z kolei o odpowiedź na pytanie: czy istnieje efektywna metoda (algorytm), który w skończonej liczbie kroków rozstrzygałby, czy dana formuła logiki pierwszego rzędu jest, czy też nie jest twierdzeniem tej logiki? To właśnie zagadnienie stało się, jak sądzą niektórzy, bezpośrednim impulsem do powstania koncepcji Churcha i Turinga[2]. Odpowiedź była negatywna — taka procedura nie istnieje. Wyraża ją twierdzenie o nierozstrzygalności:

(I) *Zbiór twierdzeń logiki pierwszego rzędu jest nierozstrzygalny.*

---

*UWAGA: Tekst został zrekonstruowany przy pomocy środków automatycznych; możliwe są więc pewne błędy, których sygnalizacja jest mile widziana (obi@opoka.org). Tekst elektroniczny posiada odrębną numerację stron.

[1]Por. w tej sprawie twierdzenie Matiasiewicza, który negatywnie odpowiedział na pytanie Hilberta, np. Z. Adamowicz, P. Zbierski *Logika matematyczna*, PWN, Warszawa 1991, ss. 194–210.

[2]Por. Y. Gurevich, The Sequential ASM Thesis, *Bulletin of European Association for Theoretical Computer Science*, February 1999,s. 3, to appear.

Ze względu na twierdzenie Gödla o pełności dla tej logiki, możemy sformułować semantyczną wersję Entscheidungsproblem[3]: czy istnieje efektywna metoda (algorytm), która w skończonej liczbie kroków rozstrzygałaby, czy dane zdanie logiki pierwszego rzędu jest, czy też nie jest tautologią tej logiki? Wobec tego mamy twierdzenie równoważne:

(II) *Zbiór tautologii logiki pierwszego rzędu jest nierozstrzygalny.*

Naszkicujemy obecnie dowód tego twierdzenia[4]. Wpierw jednak zapoznamy Czytelnika z intuicyjną ideą maszyny Turinga. Można ją sobie wyobrazić jako konkretne fizyczne urządzenie, składające się z nieskończonej (w lewo i w prawo) taśmy, podzielonej na kratki. W każdej kratce może się znaleźć jeden ze skończonej liczby symboli, które to symbole maszyna rozpoznaje. Po taśmie porusza się czytnik, urządzenie, które posiada skończoną liczbę tzw. stanów wewnętrznych. Czytnik w każdym momencie, gdy pracuje, znajduje się nad dokładnie jedną z kratek, czytając symbol wpisany w tę kratkę. Ów czytnik może wykonać jedną z następujących operacji: (1) zatrzymać się; (2) przesunąć się o jedną kratkę w lewo; (3) przesunąć się o jedną kratkę w prawo; (4) wpisać dowolny z symboli rozpoznawanych w miejsce dowolnego symbolu znajdującego się w kratce odczytywanej. Konkretną maszynę Turinga można scharakteryzować przez opis operacji, które wykonuje. Taki opis maszyny nazwiemy jej grafem. Każda operacja maszyny Turinga może być opisana w następujący sposób: (a) podać stan wewnętrzny w jakim maszyna się obecnie znajduje (obecnie, to znaczy w określonym momencie dyskretnego czasu); (b) podać symbol odczytywany w kratce, nad którą się znajduje obecnie czytnik; (c) podać operację, którą wykona maszyna (będzie to jedna z $n+3$ operacji, gdzie n jest liczbą symboli maszyny); (d) podać następny stan maszyny (po wykonaniu operacji).

Jak widać dowolną maszynę Turinga utożsamić można (w sensie rozpoznania jak działa) z jej grafem.

Powrócimy obecnie do dowodu twierdzenia (II). Podstawowym spostrzeżeniem jest to, że każdy graf dowolnej maszyny Turinga M, może być opisany za pomocą skończonego rozszerzenia L logiki pierwszego rzędu. Kluczową rolę w dowodzie pełni następujący lemat:

---

[3] J. R. Buechi, Turing–Machines and the Entscheidungsproblem, *Math. Annalen*, 148 (1962), s. 201.

[4] Dowód pochodzi istotnie od J. R. Buechi, Turing–Machines and the Entscheidungsproblem, Math. Annalen, 148 (1962), ss. 201–213. Niniejszy od G. S. Boolos, R. C. Jeffrey, *Computability and Logic*, Cambridge University Press 1990, trzecie wydanie, ss. 112–119.

(III) Ze skończonego zbioru zdań Δ wynika semantycznie zdanie H, wtedy i tylko wtedy, gdy maszyna M zatrzyma się mając na wejściu (input) k, tzn. M startuje w stanie wewnętrznym q_1z pierwszego symbolu z lewej strony nieprzerwanego ciągu k jedynek.

Wyjaśnienia wymaga zbiór Δ oraz zdanie H. Zdania zbioru Δ opisują operacje, które ma wykonać maszyna M. Aby taki opis był możliwy, musimy rozszerzyć język o symbole specyficzne. Są to, dla zamierzonej interpretacji zdań zbioru Δ, litery predykatowe, za pomocą których możemy wyrazić fakt, że maszyna w momencie czasowym t znajduje się w określonym stanie wewnętrznym nad polem x oraz iż maszyna w momencie czasowym t czyta określony symbol w kratce (polu) x. Prócz tego dysponujemy symbolem oznaczającym zero, symbolem następnika oraz symbolem oznaczającym relację mniejszości określoną w zbiorze liczb całkowitych. Ponieważ graf maszyny jest skończony, to i zbiór Δ jest skończony. W zbiorze tym są również zdania wyrażające pewne elementarne własności funkcji następnika. Zdaniem H jest alternatywa wszystkich takich zdań, z których każde mówi, przy zamierzonej interpretacji, że w grafie maszyny, dla pewnego momentu czasowego, pewnej kratki, pewnego stanu wewnętrznego oraz pewnego symbolu nie ma operacji (ruchu) dla maszyny. Jeśliby maszyna zawsze miała jakiś ruch do wykonania, to H byłoby dowolnym zdaniem fałszywym.

Z własności relacji wynikania oraz z faktu, że zbiór Δ jest skończonym rozszerzeniem logiki pierwszego rzędu, wiemy, iż wynikanie z (III) zachodzi wtedy i tylko wtedy, gdy tautologią jest implikacja, która w poprzedniku ma koniunkcję wszystkich elementów zbioru Δ, zaś w następniku zdanie H. Potrafilibyśmy zatem rozstrzygnąć o takim zdaniu, czy jest, czy też nie jest tautologią, pod warunkiem, że potrafilibyśmy o dowolnej maszynie Turinga rozstrzygnąć, czy się zatrzyma, czy też nie, startując w opisany wyżej sposób mając na wejściu k. Tego jednak nie potrafimy uczynić z tego powodu, że:

(IV) *Problem stopu (halting problem) jest nierozstrzygalny.*

Dowód tego właśnie twierdzenia zakłada, w sposób istotny, TC. Dokładnie rzecz biorąc twierdzenie (IV) jest odpowiedzią na pytanie: czy istnieje efektywna (algorytmiczna) procedura, której zastosowanie wobec dowolnej maszyny Turinga M, po skończonej ilości kroków, dawałaby odpowiedź na pytanie czy maszyna M się zatrzyma, czy też się nie zatrzyma? Załóżmy, dla dowodu niewprost, że taka procedura istnieje. Gdyby tak było, to można by obliczyć efektywnie następującą funkcję p[5]. Rozważać będziemy obecnie

[5]Można w sposób szybszy dowieść twierdzenia (IV). Niniejszy dowód pochodzi od Tibora Rado, a w tej wersji od Boolos, Jeffrey op. cit., ss. 34–41.

maszyny, które rozpoznają jedynie dwa symbole; symbol oznaczający to, że kratka jest pusta oraz symbol jedynki — 1. Niech maszyna M oblicza dowolną funkcję f określoną na całym zbiorze liczb naturalnych. Przez produktywność dowolnej takiej maszyny M, rozumieć będziemy wartość $f(0)$, tzn. liczbę jedynek jaką wydrukuje maszyna, zatrzymując się w standardowej konfiguracji, czyli nad pierwszą jedynką z lewej strony. Jeśli maszyna po zastartowaniu z pustej taśmy nie zatrzymała się lub zatrzymała w innej konfiguracji, to jej produktywność wynosi 0. Możemy teraz określić funkcję p:

(V) $p(\mathrm{n})$ = produktywność najbardziej produktywnej maszyny n–stanowej.

Funkcja ta, gdyby problem stopu był rozstrzygalny, byłaby w sensie intuicyjnym obliczalna. Oto, ponieważ istnieje tylko skończona liczba grafów dla maszyn n–stanowych, odpowiednio długo czekając, moglibyśmy doczekać się konfiguracji, w jakiej zatrzymałyby się maszyny, które się zatrzymają. Te maszyny które się nie zatrzymają, na mocy założenia potrafimy je wskazać, miałyby produktywność 0, zgodnie z definicją (V). Jeśli wykażemy nieobliczalność funkcji p, to zgodnie z prawem transpozycji, problem stopu okaże się być nierozstrzygalnym.

Załóżmy niewprost, że istnieje maszyna Turinga P k–stanowa, która oblicza funkcję p. Funkcja ta ma następujące własności:

(i) $p(1) = 1$; największa produktywność maszyny jednostanowej wynosi 1,

(ii) $p(47) \geq 100$; bo istnieje maszyna, która do wypisania dwudziestu pięciu jedynek potrzebuje dwudziestu pięciu stanów, zaś maszyna podwajająca tę liczbę ma jedenaście stanów (dołączona dwukrotnie),

(iii) $p(\mathrm{n} + 1) > p(\mathrm{n})$;

(iv) $p(\mathrm{n} + 11) \geq 2\mathrm{n}$; ponieważ maszyna podwajająca ma jedenaście stanów.

Zachodzi:

(v) $p(\mathrm{n} + 2\mathrm{k}) \geq p(p(\mathrm{n}))$.

Jest tak, ponieważ jeśli weźmiemy maszynę n–stanową, która pisze n jedynek, a do niej dołączymy maszynę P dwukrotnie, szeregowo, to taka złożona maszyna będzie miała produktywność $p(p(\mathrm{n}))$. Równocześnie taka maszyna ma $(\mathrm{n} + 2\mathrm{k})$ stanów. Zatem największa produktywność maszyny o takiej liczbie stanów jest większa lub równa jej produktywności.

Na mocy (iii) mamy:

(vi) jeśli i >j, to p(i) > p(j); stąd natychmiast: jeśli p(j) ≥ p(i), to j ≥ i, dla dowolnych naturalnych i, j.

Niech j = (n + 2k) oraz i = p(n). Mamy:

(vi) (n + 2k) ≥ p(n), dla dowolnego n.

(vii) Zatem (n + 11 + 2k) ≥ p(n + 11); z (vi),

(viii) Stąd i z (iv): (n + 11 + 2k) ≥ 2n; dla dowolnego n,

(ix) Odejmując stronami n, i podstawiając n = (12 + 2k) uzyskujemy 0 ≥ 1, sprzeczność.

Założenie o istnieniu maszyny P doprowadziło nas do absurdu, zatem taka maszyna nie istnieje. Znaczy to, że problem stopu nie jest rozstrzygalny.

Wróćmy do głównego problemu, czyli do TC. Gdzie została ona użyta? Dokładnie w dowodzie tego, że maszyna P nie istnieje. W tej jego partii, gdzie pytając o obliczalność funkcji p, założyliśmy, iż jeśli p jest obliczalna, to musi ją obliczać jakaś maszyna Turinga. Została zatem założona prawdziwość następującej implikacji:

(TC) Jeśli funkcja jest obliczalna w sensie intuicyjnym, to jest obliczalna przez jakąś maszynę Turinga.

Czy bez TC dało by się dowieść nierozstrzygalności problemu stopu? Bez TC, w powyższej postaci, owa procedura obliczająca funkcję p mogłaby spełniać warunek efektywności, ale nie wiedzielibyśmy nawet jak takiej procedury poszukiwać, ani jakie posiadałaby własności. Co za tym idzie, nierozstrzygalność logiki pierwszego rzędu nie byłaby dowiedziona, ponieważ ten problem zredukowaliśmy do rozstrzygalności problemu stopu. Jest dla mnie zagadką fakt, że ściśle matematyczne twierdzenia mogą zależeć od prawdziwości założeń, których nawet wypowiedzieć ściśle nie można. Czyżby to miało być potwierdzeniem poglądu, że logika jest w pewnym sensie nauką empiryczną? Problematyka związana z TC pokazuje również, że definicja prawdy Tarskiego nie rozwiązuje problemu prawdy dla nauk formalnych, jak się niektórym wydaje[6].

[6]Por. uwagę A. Mostowskiego zawartą w ostatnich dwóch zdaniach jego pracy *Logika matematyczna*, Warszawa–Wrocław 1948, s. 375.